

UNIVERSIDADE DO OESTE DE SANTA CATARINA – UNOESC

ADRIEL GRAUPP

LUCAS SPIER

MATHEUS CHRISTMANN

BOVINO VAX

São Miguel do Oeste

2023

ADRIEL GRAUPP

LUCAS SPIER

MATHEUS CHRISTMANN

BOVINO VAX

Trabalho apresentado ao curso de Ciência da
Computação como requisito para obtenção
de nota para o componente curricular de
Computação Aplicada a Regionalidade.

São Miguel do Oeste

2023

SUMÁRIO:

Introdução	3
BovinoVax.	4
Requisitos do Sistema	5
Histórias de Usuário.....	7
Diagramas.....	8
Scripts de Criação do banco de dados.....	9
Ferramentas Utilizadas.....	14
Desenho das Telas – utilizando o Figma.....	15
Conclusão.....	21
Referências.....	22

INTRODUÇÃO

O setor agropecuário desempenha um papel vital na economia global, fornecendo alimentos essenciais para a humanidade e contribuindo significativamente para o crescimento econômico de muitos países. Dentro desse ecossistema, o cuidado com a saúde dos animais, incluindo bovinos, é de extrema importância para garantir a qualidade da produção de carne e leite, bem como a sustentabilidade do setor. Entre os aspectos cruciais do manejo de animais, a administração de vacinas desempenha um papel central na prevenção de doenças e no aumento da produtividade.

Com o avanço da tecnologia, aplicativos móveis têm se tornado ferramentas poderosas para otimizar processos em diversas áreas, e o setor agropecuário não ficou para trás. O aplicativo BovinoVax é um exemplo notável de como a tecnologia pode ser empregada para melhorar a gestão das vacinas em bovinos e outros animais de produção. Desenvolvido utilizando o framework Flutter, o BovinoVax é uma solução inovadora e altamente eficaz para agricultores, pecuaristas e veterinários que buscam simplificar e aprimorar o controle de vacinas em seus rebanhos.

Neste trabalho, exploraremos em detalhes o aplicativo BovinoVax, destacando suas funcionalidades, benefícios e impacto no setor agropecuário. Além disso, abordaremos como essa tecnologia móvel pode contribuir para a saúde animal, a qualidade dos produtos agropecuários e, em última instância, para o sucesso e a sustentabilidade da indústria agropecuária. Através da análise do BovinoVax, esperamos demonstrar como a combinação da tecnologia e da agricultura pode resultar em soluções inovadoras que beneficiam não apenas os produtores, mas também a sociedade como um todo.

BOVINO VAX

Atualmente, a gestão de rebanhos animais enfrenta desafios significativos. Manter um controle preciso das informações de cada animal, desde detalhes individuais até históricos de saúde, é uma tarefa complexa e propensa a erros. A falta de um sistema de registro e monitoramento eficiente leva a informações desatualizadas, dificultando a tomada de decisões informadas sobre a saúde dos animais e a administração adequada de vacinações.

A ausência de um método sistemático para criar, editar e gerenciar alertas para vacinações futuras resulta em atrasos e esquecimentos, comprometendo a saúde dos animais e impactando negativamente a produtividade do rebanho. Além disso, a dificuldade em identificar cada bovino de forma rápida e precisa durante os procedimentos de vacinação contribui para ineficiências operacionais.

O sistema BovinoVax oferece uma solução abrangente para esses problemas. Ele permitirá o cadastro detalhado de informações de cada animal, desde identificação e raça até histórico de vacinações e características individuais. Isso resultará em um rebanho organizado, com informações facilmente acessíveis, fornecendo aos proprietários e veterinários uma visão clara da saúde e do estado de cada animal.

Além disso, o BovinoVax possibilitará o registro preciso das vacinações, associando cada intervenção de saúde a cada animal. Com a funcionalidade de alertas e lembretes, o sistema notificará automaticamente os usuários sobre vacinações futuras, garantindo que nenhuma data seja esquecida. A integração de tecnologias de identificação por brinco ou código de barras agilizará o processo de identificação dos bovinos durante as vacinações, aumentando a eficiência e a precisão.

Em resumo, o BovinoVax resolverá os desafios atuais de gestão de rebanho, fornecendo uma plataforma centralizada para o cadastro, monitoramento e administração de vacinações, melhorando a saúde dos bovinos e otimizando as operações agrícolas.

.

REQUISITOS DO SISTEMA

Os requisitos de um sistema são divididos em duas categorias, os funcionais e os não funcionais. Os requisitos funcionais definem o que um sistema deve fazer em termos de funcionalidades e interações, enquanto os requisitos não funcionais descrevem como o sistema deve atuar em relação a desempenho, segurança, usabilidade e outras características. Os requisitos funcionais se concentram no "o quê", enquanto os requisitos não funcionais abordam o "como" o sistema deve operar. Ambos são essenciais para garantir que o sistema atenda eficazmente às necessidades e expectativas dos usuários e do negócio. A seguir veja os requisitos de nosso sistema:

Requisitos Funcionais (RF):

RF.001 - O sistema deve permitir o Cadastro de informações detalhadas sobre cada bovino, incluindo dados como identificação do animal, raça, data de nascimento, sexo, peso e outras características relevantes.

RF.002 - O sistema deve permitir a edição do cadastro de bovinos, podendo alterar dados caso necessário.

RF.003 - O sistema deve permitir o cadastro de tipos de vacinas e suas características, bem como a edição de seus dados.

RF.004 - O sistema deve permitir o registro/vínculo de informações sobre as vacinações realizadas em cada bovino, incluindo o tipo de vacina, data da vacinação, dose administrada e o local da aplicação.

RF.005 - O sistema deve permitir o usuário realizar a leitura do brinco do bovino para identificar que o mesmo realizou a vacinação.

RF.006 - O sistema deve manter um histórico completo de todas as vacinações realizadas para cada bovino, permitindo a visualização das datas, tipos de vacinas administradas e local de aplicação.

RF.007 - O sistema deve permitir a criação, edição e exclusão de alertas/lembretes aos usuários em relação a vacinação do bovino.

RF.008 - O sistema deve enviar alertas/lembretes aos usuários quando uma vacinação estiver próxima de expirar ou for necessária.

RF.009 - O sistema deve possibilitar a geração de relatórios personalizados, como o histórico de saúde de um bovino específico ou um resumo das atividades de saúde em um determinado período.

RF.010 - O sistema deve permitir que diferentes usuários acessem o sistema com níveis de permissão diferentes, como proprietários e veterinários da propriedade.

Requisitos Não Funcionais (RNF):

RNF.001 - Brinco para rastreio e registro do animal.

RNF.002 - Controle via Código de Barras / RFID.

RNF.003 - O sistema deve fornecer respostas rápidas para consultas e operações.

RNF.004 - A interface do usuário deve ser intuitiva e fácil de navegar, com suporte para diferentes dispositivos, como smartphones, tablets e computadores.

RNF.005 - O sistema deve conter cores neutras.

RNF.006 - O sistema deve fornecer mecanismos para realizar backups regulares dos dados e permitir a restauração desses dados em caso de perda ou falhas.

RNF.007 - O sistema deve estar disponível para uso 24 horas por dia, 7 dias por semana, com uma taxa de indisponibilidade planejada não superior a 1% do tempo total por mês, exclusivamente para manutenções programadas.

RNF.008 - O sistema deve ser projetado para ser escalável, permitindo que a capacidade seja aumentada facilmente para acomodar o crescimento de registros de bovinos e dados de vacinação ao longo do tempo.

RNF.009 - O sistema deve ser compatível com os principais navegadores web (como Chrome, Firefox, Safari) e ser responsivo para funcionar adequadamente em diferentes dispositivos (computadores, tablets e smartphones).

RNF.010 - O sistema deve ser capaz de suportar o cadastro e gerenciamento de pelo menos 10.000 registros de bovinos, sem degradação perceptível de velocidade.

HISTÓRIAS DE USUÁRIO

Histórias de usuário são uma técnica na engenharia de software que descreve requisitos de forma simples, focada no usuário e compreensível. Cada história identifica quem é o usuário, o que ele deseja fazer e por que deseja fazê-lo. Elas são usadas para compreender as necessidades dos usuários, priorizar tarefas, comunicar requisitos, facilitar o desenvolvimento incremental e validar o software. Essas histórias promovem o desenvolvimento de software orientado pelo usuário e a entrega iterativa.

- História de Usuário 1: Cadastrando os Animais

"Então, como alguém que está de olho nos bovinos da fazenda, quero poder jogar todos os detalhes sobre cada bovino no sistema. Tipo, a identificação deles, qual raça são, data de nascimento, se é macho ou fêmea, o peso e outras coisas importantes. Isso vai me ajudar a manter tudo organizado."

- História de Usuário 2: Anotando as Vacinas

"Olha só, eu sou o veterinário por aqui e preciso muito conseguir anotar as vacinas que os bovinos tomam. Quero jogar as informações sobre o tipo de vacina, quando foi aplicada, a dose e onde foi aplicada. Isso vai me ajudar a ficar de olho na saúde deles e garantir que todas as vacinas necessárias estejam em dia."

- História de Usuário 3: Criando Lembretes Personalizados

"Gostaria que tivesse algum modo de criar lembretes no sistema, tipo, alertas que eu coloco lá pra me lembrar de quando é a próxima vacinação de um animal. Isso ia ser super prático e garantir que eu não perca nenhuma data importante."

- História de Usuário 4: Lendo o Brinco Pra Facilitar

"Eu queria conseguir usar o sistema pra ler o brinco dos bovinos e identificar se eles já tomaram a vacina. Assim, fica tudo mais rápido e preciso na hora de checar quem tá com as vacinas em dia."

- História de Usuário 5: Gerando Relatórios Personalizados

"Seria demais se eu conseguisse gerar uns relatórios personalizados? Tipo, um histórico de saúde detalhado de um animal em específico ou um resumo de todas as atividades

de saúde num certo período. Isso facilitaria muito na hora de tomar decisões sobre o cuidado dos animais."

DIAGRAMAS

Diagramas na engenharia de software são ferramentas visuais importantes para comunicar, analisar, projetar e documentar sistemas complexos. Eles facilitam a compreensão, simplificam a complexidade, auxiliam na tomada de decisões e melhoram a colaboração entre as equipes. Além disso, os diagramas ajudam no gerenciamento de projetos e estimativas de tempo. Em resumo, os diagramas desempenham um papel fundamental no desenvolvimento de software de qualidade. Neste trabalho utilizamos dois tipos de diagramas, diagrama de casos de uso e diagrama de Entidade Relacionamento.

Casos de Uso:

Na Linguagem de modelagem unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema. Na criação de um, use-se um conjunto de símbolos e conectores especializados. O diagrama de caso de uso tem como principais finalidade ajudar sua equipe a representar e discutir cenários e interações entre usuário e sistema. Logo abaixo você poderá conferir a interação entre usuário e sistema no projeto:

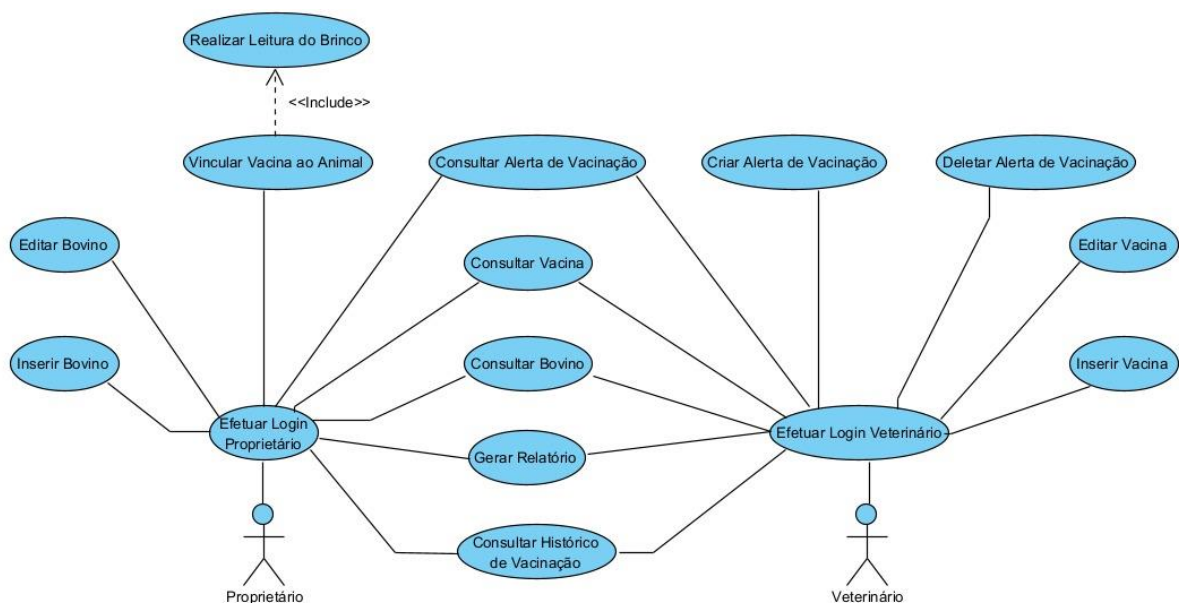
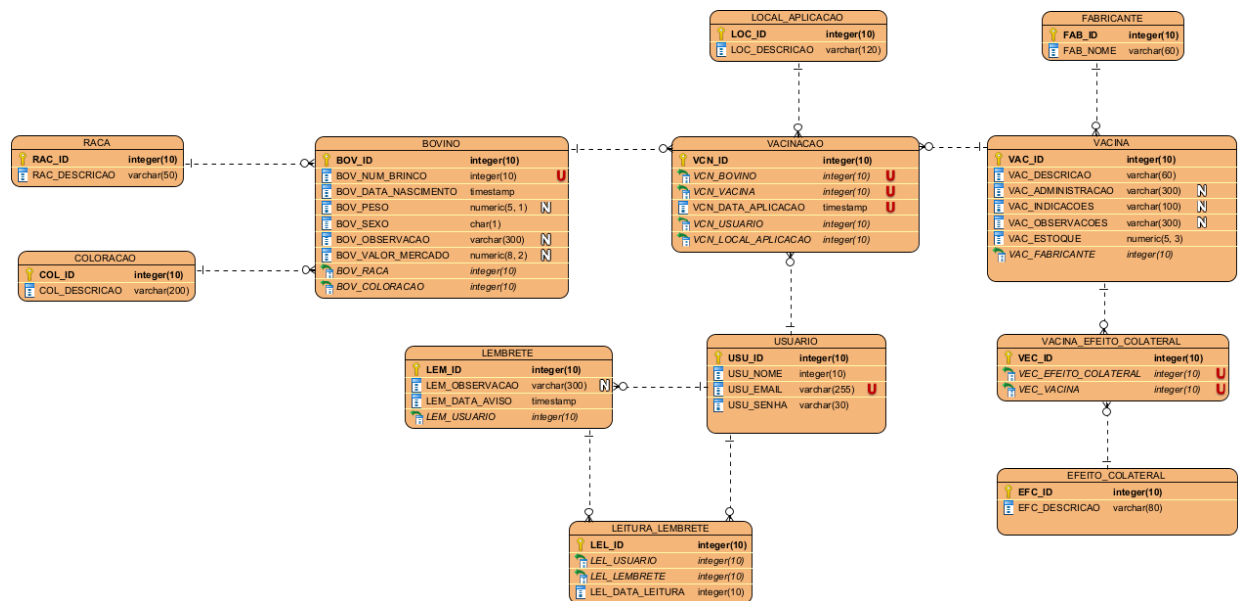


Diagrama de Entidade Relacionamento

Um diagrama de entidade-relacionamento (DER) é uma representação visual que descreve como as entidades (tabelas) em um banco de dados estão relacionadas entre si, destacando os atributos (campos) e as conexões. Usamos ele para definir estruturas de dados, comunicar requisitos, identificar chaves e restrições, e aprimorar o design do banco de dados, sendo fundamental no planejamento e modelagem de sistemas de armazenamento de dados.



SCRIPITS DE CRIAÇÃO DO BANCO DE DADOS

Scripts de criação de banco de dados são conjuntos de comandos que definem a estrutura, tabelas e relacionamentos de um banco de dados. Eles servem para garantir consistência, documentar a estrutura, facilitar implantações, backups, padronização e manutenção do banco de dados, sendo essenciais na administração e no gerenciamento eficaz dos dados.

Segue a seguir os scripts para a criação do banco de dados:

Criação das Tabelas:

```
CREATE TABLE ANIMAL(  
    ANI_ID SERIAL NOT NULL,  
    ANI_NUM_BRINCO integer NOT NULL UNIQUE,  
    ANI_DATA_NASCIMENTO timestamp NOT NULL,
```

```

    ANI_PESO numeric(5,1),
    ANI_SEXO char(1) NOT NULL,
    ANI_OBSERVACAO varchar(300),
    ANI_VALOR_MERCADO numeric(8,2),
    ANI_COLORACAO integer NOT NULL,
    ANI_RACA integer NOT NULL,
    PRIMARY KEY (ANI_ID)
);

CREATE TABLE RACA(
    RAC_ID SERIAL NOT NULL,
    RAC_DESCRICAO varchar(50) NOT NULL,
    RAC_ESPECIE integer NOT NULL,
    PRIMARY KEY (RAC_ID)
);

CREATE TABLE LEMBRETE(
    LEM_ID SERIAL NOT NULL,
    LEM_OBSERVACAO varchar(300),
    LEM_DATA_AVISO timestamp NOT NULL,
    LEM_USUARIO integer NOT NULL,
    PRIMARY KEY (LEM_ID)
);

CREATE TABLE VACINA(
    VAC_ID SERIAL NOT NULL,
    VAC_DESCRICAO varchar(60) NOT NULL,
    VAC_ADMINISTRACAO varchar(300),
    VAC_INDICACOES varchar(100),
    VAC_OBSERVACOES varchar(300),
    VAC_ESTOQUE numeric(5,3) NOT NULL,
    VAC_FABRICANTE integer NOT NULL,
    PRIMARY KEY (VAC_ID)
);

```

);

```
CREATE TABLE VACINACAO(  
    VCN_ID SERIAL NOT NULL,  
    VCN_ANIMAL integer NOT NULL UNIQUE,  
    VCN_VACINA integer NOT NULL UNIQUE,  
    VCN_DATA_APLICACAO timestamp NOT NULL UNIQUE,  
    VCN_USUARIO integer NOT NULL,  
    VCN_LOCAL_APLICACAO integer NOT NULL,  
    PRIMARY KEY (VCN_ID)  
);
```

```
CREATE TABLE FABRICANTE(  
    FAB_ID SERIAL NOT NULL,  
    FAB_NOME varchar(60) NOT NULL,  
    PRIMARY KEY (FAB_ID)  
);
```

```
CREATE TABLE USUARIO(  
    USU_ID SERIAL NOT NULL,  
    USU_NOME integer NOT NULL,  
    USU_EMAIL varchar(255) NOT NULL UNIQUE,  
    USU_SENHA varchar(30) NOT NULL,  
    PRIMARY KEY (USU_ID)  
);
```

```
CREATE TABLE COLORACAO(  
    COL_ID SERIAL NOT NULL,  
    COL_DESCRICAO varchar(200) NOT NULL,  
    PRIMARY KEY (COL_ID)  
);
```

```
CREATE TABLE LOCAL_APLICACAO(  

```

```
    LOC_ID SERIAL NOT NULL,  
    LOC_DESCRICAO varchar(120) NOT NULL,  
    PRIMARY KEY (LOC_ID)  
);
```

```
CREATE TABLE EFEITO_COLATERAL(  
    EFC_ID SERIAL NOT NULL,  
    EFC_DESCRICAO varchar(80) NOT NULL,  
    PRIMARY KEY (EFC_ID)  
);
```

```
CREATE TABLE VACINA_EFEITO_COLATERAL(  
    VEC_ID SERIAL NOT NULL,  
    VEC_EFEITO_COLATERAL integer NOT NULL UNIQUE,  
    VEC_VACINA integer NOT NULL UNIQUE,  
    PRIMARY KEY (VEC_ID)  
);
```

```
CREATE TABLE LEITURA_LEMBRETE(  
    LEL_ID SERIAL NOT NULL,  
    LEL_USUARIO integer NOT NULL,  
    LEL_LEMBRETE integer NOT NULL,  
    LEL_DATA_LEITURA integer NOT NULL,  
    PRIMARY KEY (LEL_ID)  
);
```

```
CREATE TABLE ESPECIE(  
    ESP_ID SERIAL NOT NULL,  
    ESP_DESCRICAO varchar(80) NOT NULL,  
    PRIMARY KEY (ESP_ID)  
);
```

Definindo Chaves estrangeiras:

```
ALTER TABLE VACINACAO ADD CONSTRAINT FK_VCN_ANIMAL FOREIGN KEY  
(VCN_ANIMAL) REFERENCES ANIMAL (ANI_ID);
```

```
ALTER TABLE VACINACAO ADD CONSTRAINT FK_VCN_VACINA FOREIGN KEY  
(VCN_VACINA) REFERENCES VACINA (VAC_ID);
```

```
ALTER TABLE VACINA ADD CONSTRAINT FK_VAC_FABRICANTE FOREIGN KEY  
(VAC_FABRICANTE) REFERENCES FABRICANTE (FAB_ID);
```

```
ALTER TABLE VACINACAO ADD CONSTRAINT FK_VCN_USUARIO FOREIGN  
KEY (VCN_USUARIO) REFERENCES USUARIO (USU_ID);
```

```
ALTER TABLE LEMBRETE ADD CONSTRAINT FK_LEM_USUARIO FOREIGN KEY  
(LEM_USUARIO) REFERENCES USUARIO (USU_ID);
```

```
ALTER TABLE ANIMAL ADD CONSTRAINT FK_ANI_COLORACAO FOREIGN KEY  
(ANI_COLORACAO) REFERENCES COLORACAO (COL_ID);
```

```
ALTER TABLE VACINACAO ADD CONSTRAINT FK_VCN_LOCAL_APLICACAO  
FOREIGN KEY (VCN_LOCAL_APLICACAO) REFERENCES LOCAL_APLICACAO  
(LOC_ID);
```

```
ALTER TABLE VACINA_EFEITO_COLATERAL ADD CONSTRAINT  
FK_VEC_EFEITO_COLATERAL FOREIGN KEY (VEC_EFEITO_COLATERAL)  
REFERENCES EFEITO_COLATERAL (EFC_ID);
```

```
ALTER TABLE VACINA_EFEITO_COLATERAL ADD CONSTRAINT  
FK_VEC_VACINA FOREIGN KEY (VEC_VACINA) REFERENCES VACINA  
(VAC_ID);
```

```
ALTER TABLE LEITURA_LEMBRETE ADD CONSTRAINT FK_LEL_USUARIO  
FOREIGN KEY (LEL_USUARIO) REFERENCES USUARIO (USU_ID);
```

```
ALTER TABLE LEITURA_LEMBRETE ADD CONSTRAINT FK_LEL_LEMBRETE  
FOREIGN KEY (LEL_LEMBRETE) REFERENCES LEMBRETE (LEM_ID);
```

```
ALTER TABLE RACA ADD CONSTRAINT FK_RAC_ESPECIE FOREIGN KEY  
(RAC_ESPECIE) REFERENCES ESPECIE (ESP_ID);
```

```
ALTER TABLE ANIMAL ADD CONSTRAINT FK_ANI_RACA FOREIGN KEY  
(ANI_RACA) REFERENCES RACA (RAC_ID);
```

FERRAMENTAS UTILIZADAS

O desenvolvimento do sistema BovinoVax envolve uma cuidadosa seleção de ferramentas que desempenham papéis cruciais em diferentes aspectos do projeto.

Front-End (Interface do Usuário):

- Flutter: Framework de desenvolvimento, um kit de desenvolvimento de interface de usuário
- Figma: Desenho das telas.

Back-End (Lógica de Negócios e Banco de Dados):

- Flutter: Framework de desenvolvimento, um kit de desenvolvimento de interface de usuário.
- PostgreSQL: Bancos de dados que será usado para armazenar informações de bovinos, vacinações e outros dados relacionados.

Integração de Identificação por Brinco ou Código de Barras:

- Leitores RFID: Hardware para ler brincos RFID.
- APIs ou Bibliotecas específicas: Para processar os dados lidos dos dispositivos de identificação.

Hospedagem e Implantação:

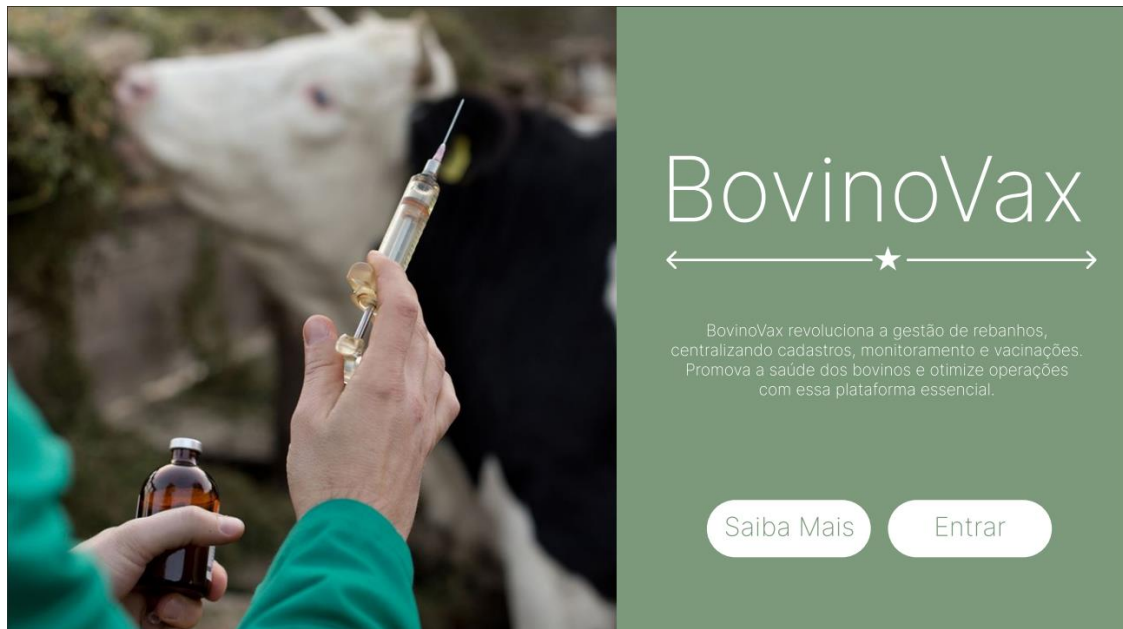
- AWS: Plataforma de hospedagem em nuvem para implantar seu aplicativo.
- GitHub: hospedar o app durante seu desenvolvimento

DESENHO DAS TELAS – UTILIZANDO O FIGMA

O Figma é uma ferramenta de design e prototipagem baseada na web amplamente utilizada, conhecida por sua colaboração em tempo real, multiplataforma e facilidade de uso. Permite que designers e equipes criem interfaces de usuário visualmente atraentes, desenvolvam protótipos interativos, compartilhem projetos facilmente e garantam a consistência do design por meio de bibliotecas de componentes. Sua acessibilidade, segurança e integrações com outras ferramentas o tornam uma escolha popular para criar e colaborar em projetos de design de interface de usuário. Segue o desenho das telas de nosso projeto:

Home:

Tela inicial do sistema, que mostra uma breve apresentação do nosso app e as opções de login e de acessar a tela de saiba mais.



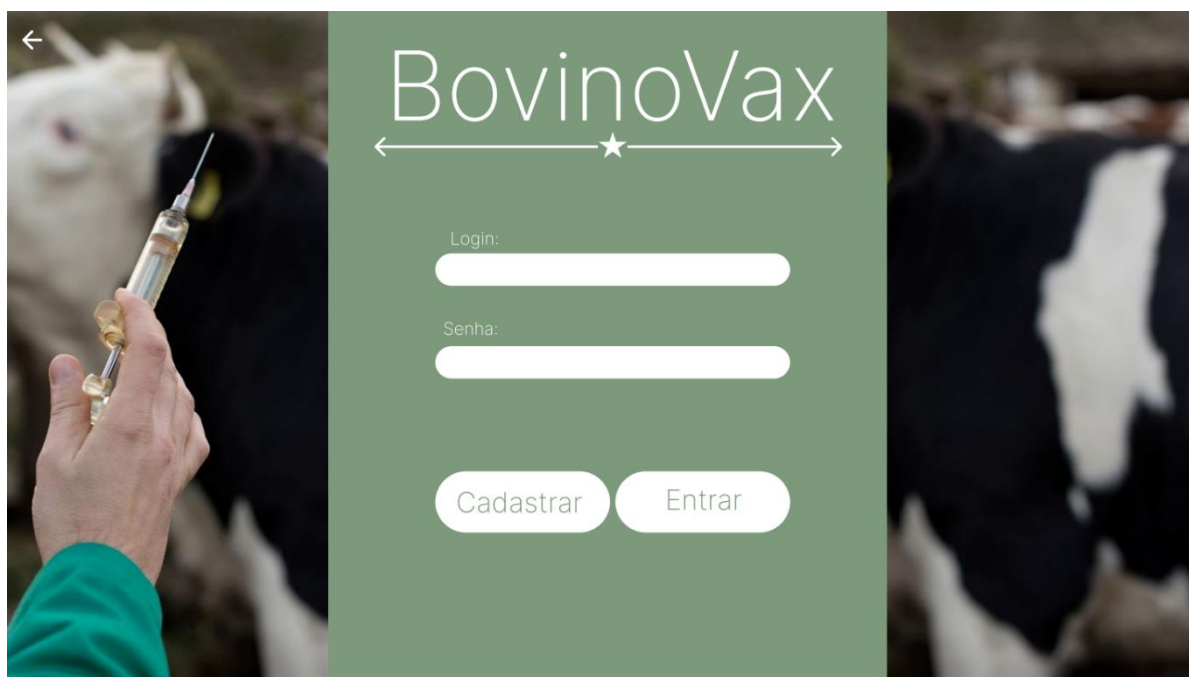
Saiba Mais:

Tem um texto explicativo da ideia do projeto, qual é o seu objetivo, o que ele pode oferecer...



Login:

Tela onde usuário irá usar para logar no sistema, ali ele informa seu login e senha



Lembretes:

Tela onde será exibido todos os lembretes de vacinação nos animais. Mostrando a data que deve ser feito a vacina, qual o bovino a ser vacinado, qual é a vacina a ser aplicada e o local de aplicação do medicamento.

BovinoVax

@Usuario123

LembretesCadastro de BovinoVacinarCadastro de VacinasRelatórios

Próximos Eventos:

BuscarNovo Lembrete

Hoje:

Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

Proximos Dias:

21/08
Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

21/08
Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

22/08
Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

24/08
Vacinar Bovino: XXXXXXXX
Tipo da Vacina: XXXXXXXX
Local: XXXXXXXX
Raça: XXXXXXXX

Cadastro de Lembretes:

Tela usada para fazer o cadastro dos lembretes para vacinar o animal, onde o usuário informa o animal, a vacina e a data que será feito a vacina no animal.

BovinoVax

@Usuario123

LembretesCadastro de BovinoVacinarCadastro de VacinasRelatórios

Novo Lembrete

Numero do Brinco:

Nome da Vacina:

Data de vacinação:

Cadastro de Animal:

Tela utilizada para realizar o cadastro de um novo animal no rebanho., onde é possível salvar várias características do animal

BovinoVax @Usuario123

Lembretes **Cadastro de Bovino** Vacinar Cadastro de Vacinas Relatórios

Cadastro de Bovino

Numero do Brinco: **Buscar**

Raça:

Data de Nascimento:

Peso: Cor:

Valor: Sexo: ☒ Macho ☐ Fêmea

Observações:

Histórico de Vacinações:

23/11/2022 - Vacina XXXXX

Lembretes:

21/08/2023 - Vacina XXXXX

04/05/2024 - Vacina XXXXX

Novo **Editar** **Remover**

Salvar **Cancelar**

Tela de Vacinar:

Tela utilizada para registrar o processo de vacinação em uma animal, onde é possível selecionar o animal, a vacina, o local de aplicação e até mesmo vincular com algum lembrete já existente, assim automaticamente já puxando a vacina e outros dados que foram cadastrados no lembrete.

BovinoVax @Usuario123

Lembretes Cadastro de Bovino **Vacinar** Cadastro de Vacinas Relatórios

Vacinar

Numero do Brinco: **Buscar**

Nome da Vacina:

Data da vacinação:

Local de aplicação:

Histórico de Vacinações:

23/11/2022 - Vacina XXXXX

Lembretes:

21/08/2023 - Vacina XXXXX

04/05/2024 - Vacina XXXXX

Selecionar

Salvar **Cancelar**

Cadastro de Vacinas:

Tela de cadastro de vacinas, onde é possível salvar alguns dados, como por exemplo: Código, nome descrição, fabricante, modo de administrar, estoque, indicações e até mesmo efeitos colaterais

The screenshot shows the 'Cadastro de Vacinas' form within the BovinoVax application. The header bar is green with the BovinoVax logo and a user profile icon labeled '@Usuario123'. A navigation bar below the header contains links: 'Lembretes', 'Cadastro de Bovino', 'Vacinar', 'Cadastro de Vacinas' (highlighted), and 'Relatórios'. The main content area is titled 'Cadastro de Vacinas' and contains a search bar labeled 'Codigo/Tipo:' with a 'Buscar' button. Below this are several input fields arranged in two columns: 'Codigo:', 'Nome:', 'Descrição:', 'Fabricante:', 'Modo de Administrar:', 'Estoque:', 'Indicações:', and 'Efeitos Colaterias:'.


Gerador de Relatórios:

Nesta tela é possível gerar relatórios referente aos eventos acontecidos em um determinado período de tempo e até sendo possível filtrar por um animal específico

The screenshot shows the 'Gerador de relatórios' form within the BovinoVax application. The header bar is green with the BovinoVax logo and a user profile icon labeled '@Usuario123'. A navigation bar below the header contains links: 'Lembretes', 'Cadastro de Bovino', 'Vacinar', 'Cadastro de Vacinas', and 'Relatórios' (highlighted). The main content area is titled 'Gerador de relatórios' and contains a section labeled 'Filtros:'. Below this are input fields for 'Numero do Brinco:', 'Data inicial:', and 'Data final:'. At the bottom of the form are two buttons: 'Gerar' and 'Cancelar'.

Usuário

Nesta tela é possível alterar as características da conta do usuário

 BovinoVax 



@usuario123

Nome de Usuário:

Nome Completo:

E-mail:

Telefone:

Senha:

Nível de permissão: 

Não é possível alterar aqui

Salvar

Cancelar

CONCLUSÃO

Em conclusão, o desenvolvimento e implementação do BovinoVax representam um passo significativo em direção à melhoria da gestão da saúde dos animais na indústria agropecuária. Este sistema, criado com base no framework Flutter e integrando recursos como PostgreSQL, leitores RFID e APIs específicas, oferece uma solução abrangente e inovadora para o controle de vacinas em bovinos e outros animais de produção.

Através da combinação de tecnologia avançada com as melhores práticas de saúde animal, o BovinoVax visa aumentar a eficiência das operações, melhorar a rastreabilidade, minimizar os riscos de doenças e garantir a qualidade dos produtos agropecuários. Além disso, sua interface de usuário amigável, desenvolvida com a ajuda do Figma, simplifica a adoção e utilização por parte dos produtores e veterinários.

À medida que enfrentamos desafios globais relacionados à segurança alimentar e à sustentabilidade, soluções como o BovinoVax desempenham um papel crucial na otimização da produção agropecuária. Este trabalho ilustra como a tecnologia pode ser aplicada de forma eficaz para aprimorar a gestão da saúde animal, promovendo o bem-estar dos animais e contribuindo para o sucesso contínuo da indústria agropecuária. Com sua implementação e a contínua colaboração entre especialistas em tecnologia e profissionais do setor agropecuário, o BovinoVax tem o potencial de beneficiar significativamente a sociedade como um todo, ao garantir a oferta de produtos agropecuários seguros e saudáveis.

REFERÊNCIAS:

OPENAI. ChatGPT. Disponível em: <<https://chat.openai.com/>>.

FIGMA. Figma: the collaborative interface design tool. Disponível em: <<https://www.figma.com/>>.